GIS MULTIMEDIA TEACHING TOOL ABOUT AFRICA

_____

A Thesis

Presented to the

Faculty of

San Diego State University

_____

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

_____

by

Vivek S. Shah

Spring 2012

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Vivek S. Shah:

GIS Multimedia Teaching Tool about Africa

Carl Eckberg, Chair
Department of Computer Science

Joseph Lewis
Department of Computer Science

Steve Kirschvink
Department of Mathematics

Nov 16, 2011
Approval Date

Copyright © 2012
by
Vivek S. Shah
All Rights Reserved

DEDICATION

I dedicate this thesis to my family, for their encouragement and their unconditional love which they bestowed upon me every time I needed it, and their belief that one day I could make an achievement and to all my dear friends for their wonderful support during this entire tenure and for the times to come.

ABSTRACT OF THE THESIS

GIS Multimedia Teaching Tool about Africa
by
Vivek S. Shah
Master of Science in Computer Science
San Diego State University, 2012

The motivation of this software is to develop an interactive GIS application which should enhance student activity and interest in learning world history. Computer aided teaching enables teachers to engage the students better by increasing desire and motivation to learn. This thesis focuses on creating a GIS multimedia teaching tool about the Africa continent. The tool provides interactive graphical information of colonial history, independence history, points of interests, demographics and physiography of Africa. Students can switch between various European countries (Belgian, French or UK) to see which countries were colonized when and by whom and when students use the hotlink tool on a particular country, a pop up window describes the country name, capital city, independence date, population, area, and government type. If the user wants to know more about a particular country, a Wikipedia link is also provided. Hotlink tool works the same way for the points of interest layer. Students can also view deserts and rivers of Africa. A dynamic user interface created using JAVA and the MOJO software package has been included in this tool for effective learning. Typical user interface features are provided in this tool such as zoom-in, zoom-out, a legend editor, location identifier, print command, map tips, measure, and query builder. GIS technologies like MapObjects - JAVA edition and j2sdk were used along with the Eclipse IDE to develop this tool. The tool can run standalone and the user just needs to have JRE install on his system.

TABLE OF CONTENTS

PAGE

ABSTRACT v
LIST OF TABLES viii
LIST OF FIGURES ix
ACKNOWLEDGEMENTS xi
CHAPTER
1 INTRODUCTION 1
2 TECHNOLOGY 3
2.1 Java 3
2.2 MapObjects Java Edition 5
2.3 DBF Explorer 6
2.4 Eclipse IDE 6
3 THE REQUIREMENTS 8
3.1 Data Requirements 8
3.2 Platform Requirements 8
3.3 Functional Requirements 9
4 PROTOTYPING 11
5 SOFTWARE ARCHITECTURE 14
5.1 High Level Architecture Diagram 14
5.2 Classes and Class Diagram 15
6 ECLIPSE AND MAPOBJECTS CONFIGURATION 16
7 MAPOBJECTS TOOLBAR 22
7.1 Project Toolbar 22
7.2 ZoomPanToolbar 22
7.3 Selection Toolbar 23
8 CUSTOM TOOLBAR 24
9 ADDITIONAL TOOLS 27
9.1 Hotlink 27

9.2 Legend Editor....28
10 SCREENSHOTS....31
11 CONCLUSION AND OBSTACLES....37
12 FUTURE ENHANCEMENTS....38
BIBLIOGRAPHY....39

LIST OF TABLES

PAGE

Table 5.1. Imported Classes Used by this Application 15
Table 8.1. Legend of Custom Toolbar 24

LIST OF FIGURES

PAGE

Figure 2.1. Characteristics of Java. 3
Figure 4.1. Prototype model. 11
Figure 4.2. GIS multimedia teaching tool about Africa main screen. 13
Figure 5.1. High level architecture diagram. 14
Figure 6.1. Starting Eclipse by double clicking on Eclipse icon. 17
Figure 6.2. Opening the Eclipse workbench. 17
Figure 6.3. Creating a new Java project. 18
Figure 6.4. Selecting user library. 18
Figure 6.5. Creating a new library. 19
Figure 6.6. Adding library name. 19
Figure 6.7. Selecting JAR files. 20
Figure 6.8. Make sure the new library checkbox is selected. 20
Figure 6.9. Specify the JavaDoc location. 21
Figure 6.10. Attaching JavaDoc to the JAR files for references in the JAVA editor. 21
Figure 7.1. MapObjects toolbar. 22
Figure 8.1. Print dialog. 25
Figure 8.2. Add layer dialog window. 25
Figure 9.1. GIS tool with Hotlink window. 28
Figure 9.2. Layer properties window. 29
Figure 9.3. GIS tool with graduated symbols on the layer using legend editor. 30
Figure 10.1. GIS multimedia teaching tool about Africa start screen. 31
Figure 10.2. Add layer. 32
Figure 10.3. Remove layer. 32
Figure 10.4. Identify tool. 33
Figure 10.5. ZoomToActiveLayer. 33
Figure 10.6. Hotlink tool applied on africa1 layer. 34
Figure 10.7. Wikipedia page of the country selected by Hotlink tool. 34

Figure 10.8. Hotlink tool applied on africapoi layer.....35
Figure 10.9. Map tips feature applied on NAME field africapoi layer.....36

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Carl Eckberg for all the help, support and continuous guidance received for this project.

I am grateful to express my sincere thanks to professor Joseph Lewis and Professor Steve Kirschvink for being on my thesis committee and for their help and co-operation.

CHAPTER 1

INTRODUCTION

Technology is entering every area and is providing advantages for representing things in an easily understandable way, and increasing access to information. Teaching is a good example of this. using new technology can provide a better way for students to learn dry subjects, and when web hosted, or iPhone accessible, permit learning from any location.

The teaching culture has been changing over the past decade and use of electronic media is increasing. Electronic teaching has been one of the most important mediums adapted since the increase in the use of computers and the internet. Students can be more receptive when taught via a visual and entertaining medium than through the more usual method of book reading. Many students are more inclined to newer technologies and are have an appetite to learn using them. The ability to learn while on the move is very important, for example learning a new language. Math is much harder to learn online, and the fields of history and geography, in focus here, are somewhere in between.

The purpose of this GIS multimedia teaching tool about the African continent is to provide interactive graphical information of colonial history, independence history, points of interests, demographics and physiographic of Africa. There are 54 countries in the African continent. Users can switch between various colonizers (Belgian, French or UK) of Africa to see which countries experienced colonization. When users use the hotlink tool on particular country, a pop up window gives description about the country name, capital city, independence date, population, area, and government type. If user wants to know more about a particular country, a Wikipedia link is also provided. The hotlink tool works the same way for the points of interest layer. Users can also view the deserts and rivers of Africa. Typical interactive map tools, like zoom-in are provided.

ESRI (Environmental Science Research Institute) has an outstanding presence in the field of GIS (Geographic Information Science). MapObjects is a suite of Java-based developer components for creating client or server-side mapping and GIS applications [1]. The release of MapObjects Java Standard Edition allows a developer to build custom map

applications using the open standard Java programming language. Software developers can extend the basic map functionality provide customized capabilities to satisfy the end user.

This tool was developed using MapObjects Java Objects (MOJO) technology which is a product of ESRI. The IDE used for developing this tool is Eclipse 3.5. MapObjects Java Edition is used as it can perform activities as labeling map features, thematic mapping, panning and zooming through multiple map layers, querying spatial and attribute data, performing geometric operations, measuring distances, displaying real-time geographic data, and much more. It provides features by which geographic-based display can be implemented in designing applications. MapObjects supports data types include shape files, layers, image formats such as BMP, TIFF, PNG, and JPG. The term MOJO is a non-ESRI term for MapObjects which is more easily remembered. The two great virtues of these technologies are:

1. The GIS application can be customized to a very high degree.
2. Deployment to a wide variety of platforms is made easy by Java executable jar files.

CHAPTER 2

TECHNOLOGY

This chapter focuses on the technology used for developing this tool. This tool has been developed using Java, and MapObjects Java Edition. The reason for choosing Java as the programming language is mainly because it supports MapObjects Java Edition and it is very easy to use and deploy. Both technologies will be discussed at length. In this chapter we will focus on the benefits of the technologies used and their key areas.

2.1 JAVA

Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications [2]. Figure 2.1 describes the characteristics of Java [3].

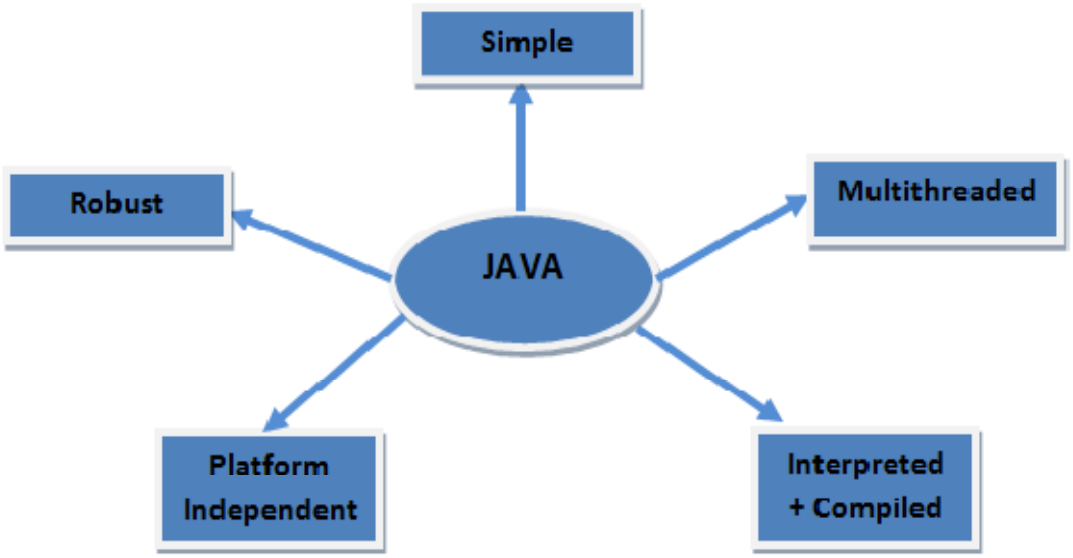

Figure 2.1. Characteristics of Java.

Following are some advantages of Java [4]:

Java is simple: Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn compared to other programming languages. One reason why Java is much simpler than C++ is because Java uses automatic memory allocation and garbage collection whereas C++ requires the programmer to allocate memory and to collect garbage himself.

Java is object-oriented: Java is object-oriented because programming in Java is centered on creating objects, manipulating objects, and making objects work together. This allows you to create modular programs and reusable code. Java is platform-independent: one of the most significant advantages of Java is its ability to move easily from one computer system to another.

Java is distributed: Distributed computing involves several computers on a network working together. Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it.

Java is interpreted: An interpreter is needed in order to run Java programs. The programs are compiled into Java Virtual Machine code called byte code. Thus Java can work without writing new compilers to work with new machines.

Java is secure: Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

Java is robust: Robust means extensible, without great risk to its reliability. Java puts a lot of emphasis on mandatory early checking for possible errors, as Java compilers force the detection of many problems that would first show up during execution time in other languages.

Java is multithreaded: Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated into it, while in other languages, operating system-specific procedures have to be called in order to enable multithreading. Multithreading is a necessity in visual and network programming.

Java is easily deployed: Java applications can be easily deployed by using executable java archives, i.e. jar files.

2.2 MAPOBJECTS JAVA EDITION

MapObjects Java Edition can be used to build custom applications that incorporate GIS and mapping capabilities or to extend the capabilities of existing applications. MapObjects Java Edition is a collection of client and server side components used to build custom, cross platform mapping and GIS applications. It allows precise control over the behavior of the application; a user can make his application from the bottom up, adding only those features that he requires; a GIS class at San Diego State University covers MapObjects in depth and has a published guide named Notes On MapObjects Java Edition by Dr. Carl Eckberg, which was one of the main criteria for choosing MapObjects [5]. Important features as cited from the MapObjects Java Edition brochure are the following [6]:

- Applications which are built using MapObjects Java Edition can support activities such as labeling map features, thematic mapping, panning and zooming through multiple map layers, specifying projections, querying spatial and attribute data, performing geometric operations, measuring distances, displaying real time geographic data, and creating layouts, creating shape files, projecting on particular locations.
- One can easily distribute MapObjects Java Edition applications over an internet or intranet through browser hosted applets or simplify web delivery of your application with the use of Sun Java web start technology, the industry standard launching mechanism for distributing Java applications over the web; one can use custom created web pages to display the information related to the topics.
- By using the rich Swing components included with MapObjects Java Edition, you can quickly build applications that include functional toolbars, dynamic symbol control, query dialogs, overview and insert maps, and intelligent legends, Pop up windows, alert boxes, java panes which have ability to display multiple format of data that make your custom applications easy to use and even easier to develop and easy to understand.
- The different Map Object packages contains many pre-executed classes, which are used to create various tools, this makes a developer’s job very easy.
- The server side map components in MapObjects Java Edition allow developers to build map services, Java Server Pages (JSPs) and servlets, or custom Enterprise JavaBeans (EJB) Web based mapping applications. Several extensive sample applications based on JSPs and EJBs are provided to demonstrate how to build robust Web applications with the server side MapObjects.
- By using applications created with MapObjects Java Edition, end users can combine local data with Internet and Intranet data to create their own customized maps and easily integrate these maps with different locations corresponding to the map projected using comma separated value files which is having latitude and longitudes

of the locations; it also supports different data formats including shape files, ArcSDE layers, ArcIMS image and feature services, image formats such as BMP, TIFF, PNG, JPG, GIF. Generally these image formats comes into the picture when we try to create different toolbars, having their own images to recognize, You can also use MapObjects Java Edition to access files from your own custom data sources for easy integration.

Other Java based GIS packaged include UDIG and JUMP, but MOJO was favored since we have a lot of developed features there already. Larger and high priced ESRI tools were not considered because, in addition to the cost issues, they are not as readily customized and deployed.

2.3 DBF EXPLORER

DBFExplorer is an easy-to-use, simple GUI based application for editing and viewing DBF files. We can easily add, edit, delete records and modify table structure by modifying the data length and data type. In our application it is used to store the data related to the shape files in a tabular structure. Some features of this tool are:

- It does not require any external drivers for database connections.
- It enables you to print and export data to HTML format.

The only experienced drawback seems to be the ESRI shapefiles resent having a column name changed [7].

2.4 ECLIPSE IDE

An Integration Development Environment plays an important role in effective application development. The important features, which IDE provides, are:

- Code editing capability which provides you IntelliSense with code refactoring. IntelliSense corresponds to the accessing the descriptions of the functions, their parameter list and it reduces the reference to the external documentation and hence speeding up the development time.
- Debugger: It helps you in debugging your code by adding the watch to the variables in the code, which helps you in monitoring the different values for the variables at different stages during the code execution. It also helps you in putting breakpoints in between the code execution so that it becomes easier to find bugs using tools like JUnit and Versioning.

I have used Eclipse as the IDE for our application. But for making eclipse available for the MapObjects Java Objects, we configured it which will be covered in detail in Chapter 6.

CHAPTER 3

THE REQUIREMENTS

The GIS Multimedia Teaching Tool about Africa is an interactive tool developed for users to enable them to learn more about the colonial history, independence history, demographics and physiography of Africa. The thesis has been developed under the guidance of Dr. Carl Eckberg from the San Diego State University Computer Science department and he was the major contributor of the requirements for this thesis.

The requirements gathered have been classified into data requirements, platform requirements and functional requirements.

3.1 DATA REQUIREMENTS

The data requirements were gathered from Prof. Carl Eckberg in order to have accurate information pertaining to the African history. He helped in identifying important information about African colonies and points of interests to be displayed for African countries. How the display of data is reflected in the tool is described in the functional requirements.

I have gathered all the information of colonial history, independence history, demographics, deserts and rivers for each African country. Each African country will cover the following information: Name of Country, Capital city, Population, Area, Colony of, Government and Independence date.

3.2 PLATFORM REQUIREMENTS

The platform requirements refer to the OS and the technologies used for developing the tool. Although this tool is platform independent, the implementation was done using the Windows platform. The platform requirements were gathered from Dr. Carl Eckberg. Java was used as the development language with MapObjects Java Edition as the GIS solution. These technologies offer a wide range of mapping and GIS functionality. MapObjects Java Edition helps build applications that perform a variety of geography-based display, query,

and data retrieval activities at the client, presentation, Web, and server tiers. Thus they allow representing the geographical data on maps which make the tool more interactive and informative [5].

3.3 FUNCTIONAL REQUIREMENTS

The functional requirements mainly include the Graphical User Interface (GUI) requirements which a user would use to interact with the tool. The requirements were gathered from Dr. Carl Eckberg. Following are the functional requirements for the GIS multimedia teaching tool for Africa [3]:

- The tool should run on any platform.
- All the data should be represented in geographical form on a map so that users can identify different countries easily.
- Tool should be loaded with all the layers.
- Different colonies should have different colors on the map so that identification can be made easy.
- All the graphical features and fonts should be neat and clean.
- The tool shall provide tools to zoom in and zoom out and it should be capable of adding more layers to the map.
- The software should be able to add and remove labeling on each layer.
- Hotlinks is an important requirement in the project. After selecting the hotlink tool and clicking on any African country, it should open a window giving information about the selected country and a Wikipedia hyperlink to the country's page.
- Hyperlink should allow the user to go to external webpage to get more information about the country.
- Hotlink and Hyperlink should work for the point of interests layer as well.
- Tool shall provide a built-in Print Option.
- Java should be used, which is simple and platform independent, so it can be utilized everywhere.
- This tool shall be created with the use of MapObjects – Java edition.
- The project should be nicely packaged in a JAR file and placed on a campus website so it can be easily downloaded and installed on any machine.

Africa has sometimes been known as "the dark continent" because it was colonized and developed after other parts of the world. It remains relatively undeveloped, and its history and geography as not at all well known. Projects similar to this one, located

elsewhere, were sought by various on campus departments, but an enthusiastic campus customer for this one was harder to find.

CHAPTER 4

PROTOTYPING

The best way to make sure that the requirements have been understood correctly, and both the developer, and the client are on the same page, is to prototype the product and demo it to the client. The prototype also provides a visual feedback to the client that can be used to generate more ideas and understand how their requirements would be implemented.

Prototyping the development cycle of your project helps in reduces the risk during implementation and adaption of the software tool. See Figure 4.1 for the steps followed in the Prototype Model. This model increases the flexibility of the development process by allowing the client to interact and experiment with a working representation of the product. The main focus of this model is to satisfy the client needs, and the developmental process only continues once the requirements are satisfied with the functioning of the prototype.

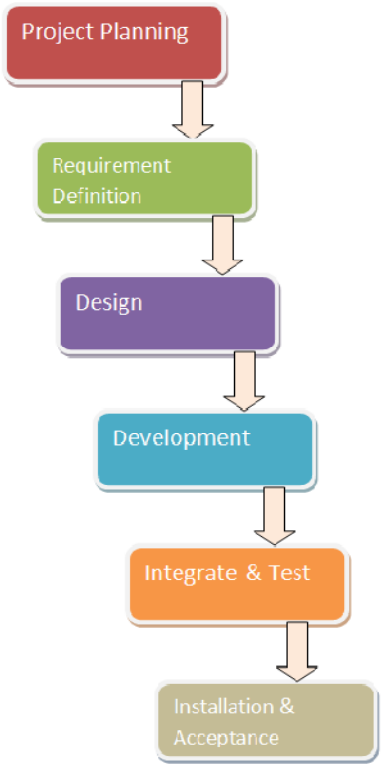

Figure 4.1. Prototype model.

The software development life cycle can be subdivided into four phases.

- Definition and Requirement Collection: Understanding requirements plays a crucial role in creating any software product. A clear definition of requirements is essential in designing the tool. In the previous chapter we already mentioned the requirements gathered. After this phase, we would have a clear definition and image of the requirements needed to develop this Multimedia GIS tool for Africa.
- Design: In this phase, using the specifications provided we can design the project; the functionalities and GUI the user expects will help in designing the outer layer of the project. It also includes the implementation and testing of software code which is ready for deployment.
- Development: This phase consists of developing the actual tool by writing code, implementing tool as per the requirements gathered. Initially small modules of different functions are implemented and then these small units are integrated to form a system. This process goes on until the project is ready to deliver. I found this process very helpful to keep track of the requirements and implementation of the project. Testing the modules at each step of implementation enhances less error rate and is easy to keep track of the actual requirements.
- Deployment and Maintenance: The actual deployment starts after the code is appropriately tested and all other aspects of the project including documentation, has been taken care of. Documentation and writing user manual is an important part in this phase.

Figure 4.2 shows the screen shots of the prototype. The screen is the main screen which is launched when the tool is started.

Initial prototype has been developed and modified according to feedback from the thesis supervisor. All the features were implemented accordingly and were finished in the time frame given as per requirements. Feedback from fellow students also played a role. The inability to find an on campus “customer” from, e.g., history, meant the absence of a local expert on Africa.

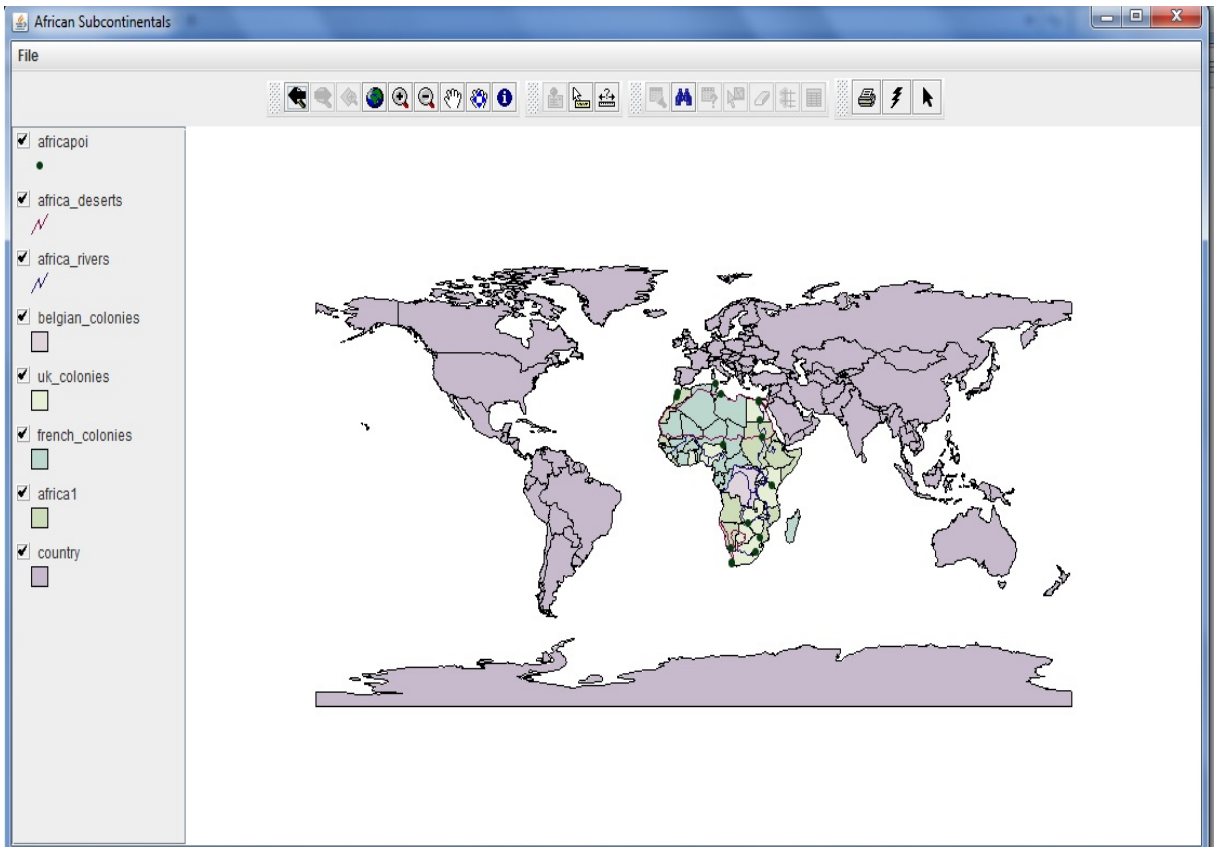

Figure 4.2. GIS multimedia teaching tool about Africa main screen.

CHAPTER 5

SOFTWARE ARCHITECTURE

The design phase started after finishing requirement collection for the tool. This chapter mainly focuses on the high level software architecture and class diagrams. The high level architecture diagram states the relationship between different entities.

5.1 HIGH LEVEL ARCHITECTURE DIAGRAM

Figure 5.1 depicts the high level architecture diagram for GIS multimedia teaching tool about Africa.

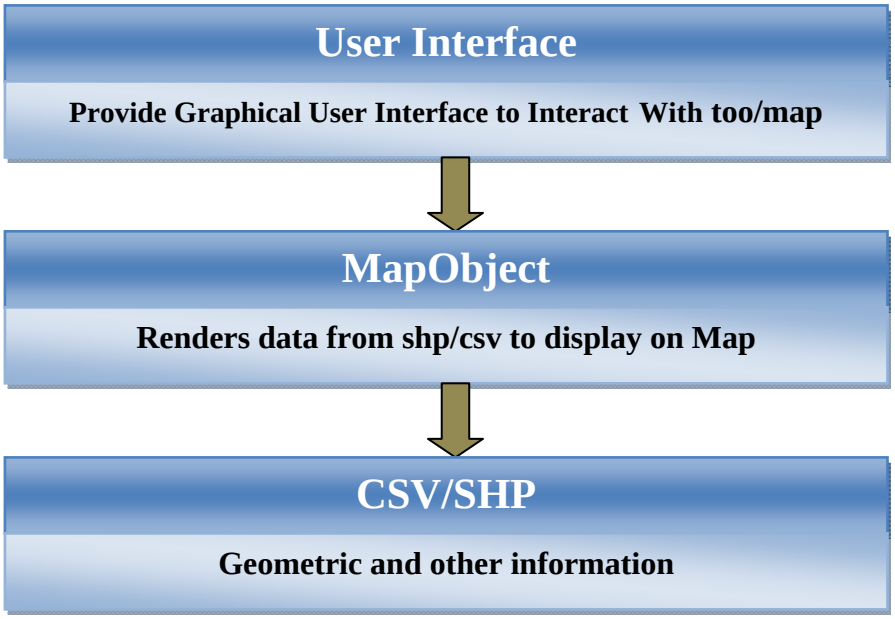
User Interface
Provide Graphical User Interface to Interact With too/map
MapObject
Renders data from shp/csv to display on Map
CSV/SHP
Geometric and other information

Figure 5.1. High level architecture diagram.

The architecture is broadly divided into three different levels:

- User Interface: It is the end user display which is used to interact with maps, and user functionality which has been developed as per requirements. It will be launched with the application start up.
- MapObjects and Java Classes: The functionality, attached with toolbars, buttons, menus and other items on the main interface, is created and defined in different JAVA classes.

- CSV/SHP: Shape files and CSV files contains geographical attributes which need to be displayed on UI by MOJO and JAVA classes. MOJO uses debase format(dbf) for non-location attributes.

5.2 CLASSES AND CLASS DIAGRAM

The class diagram displays the general organization of the system. It helps to understand how different classes are interacting with each other. A Class Diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes [8]. It also helps to understand the interdependency of classes used for development.

The classes as shown in Table 5.1 have been used to develop the functionality provided by this tool. In this project a couple of packages have been included to implement functionality provided by inbuilt classes. The MOJO classes in Table 5.1 are the JAR files.

Table 5.1. Imported Classes Used by this Application

Java Components	Java Class
import javax.swing	JMenuBar, JMenuItem, AbstractTableModel, TableColumn, JButtons, JPanel, JSplitPane, JToolBar, ImageIcon
import com.esri.mo2.ui.bean	TocAdapter, TocEvent, Map, Layer, Toc, Legend, Tool
import com.esri.mo2.ui.tb	ProjectToolBar, ZoomPanToolBar, SelectionToolBar
import com.esri.mo2.map.dpy	BaseFeatureLayer, Layerset, FeatureLayer

CHAPTER 6

ECLIPSE AND MAPOBJECTS CONFIGURATION

Eclipse IDE (Integrated Development Environment) is a multi-language software development environment comprising of an integrated development environment and plug-in system. Eclipse IDE is used for the editing, compiling and development of the GIS tool as it provides an ability to integrate with the MapObjects Java Edition beans allowing a drag and drop development. The steps that follow explain how Eclipse IDE and MapObjects are configured:

1. Install Eclipse IDE from Eclipse website.
2. Install MapObjects Java Edition installation CD from Dr. Carl Eckberg [5].
3. Install Java (SE) development kit 2 from Java site.
4. Launch Eclipse IDE from Start->Programs->Eclipse or double clicking the eclipse icon from the installation directory. Figure 6.1 the process for the Step 4.
5. Open workbench by double clicking the yellow highlighted icon. Figure 6.2 shows the process for this step.
6. Create a new Java Project in workspace. Go to File->New->Java Project to open the project wizard. Name the Project and click Finish. Figure 6.3 shows the process for this step.
7. Create a new library by right clicking the project; select Build Path ->Add Libraries It will open a window as shown in Figure 6.4.
8. Click Next and select User libraries button This will open a preferences window Click new and enter a name for the library Figure 6.5 shows the window which will open while performing this step.
9. Type a new user library name and click ok. The step execution is shown in Figure 6.6.
10. Add jar files to the newly created library by clicking Add Jars button in the preferences window.
11. Navigate to the folder where MapObjects is installed.
12. Select all the JAR files and click ok. Figure 6.7 shows the list of JAR files to select.
13. Check if the newly created library checkbox is selected. Figure 6.8 shows the step execution.
14. Click Finish.

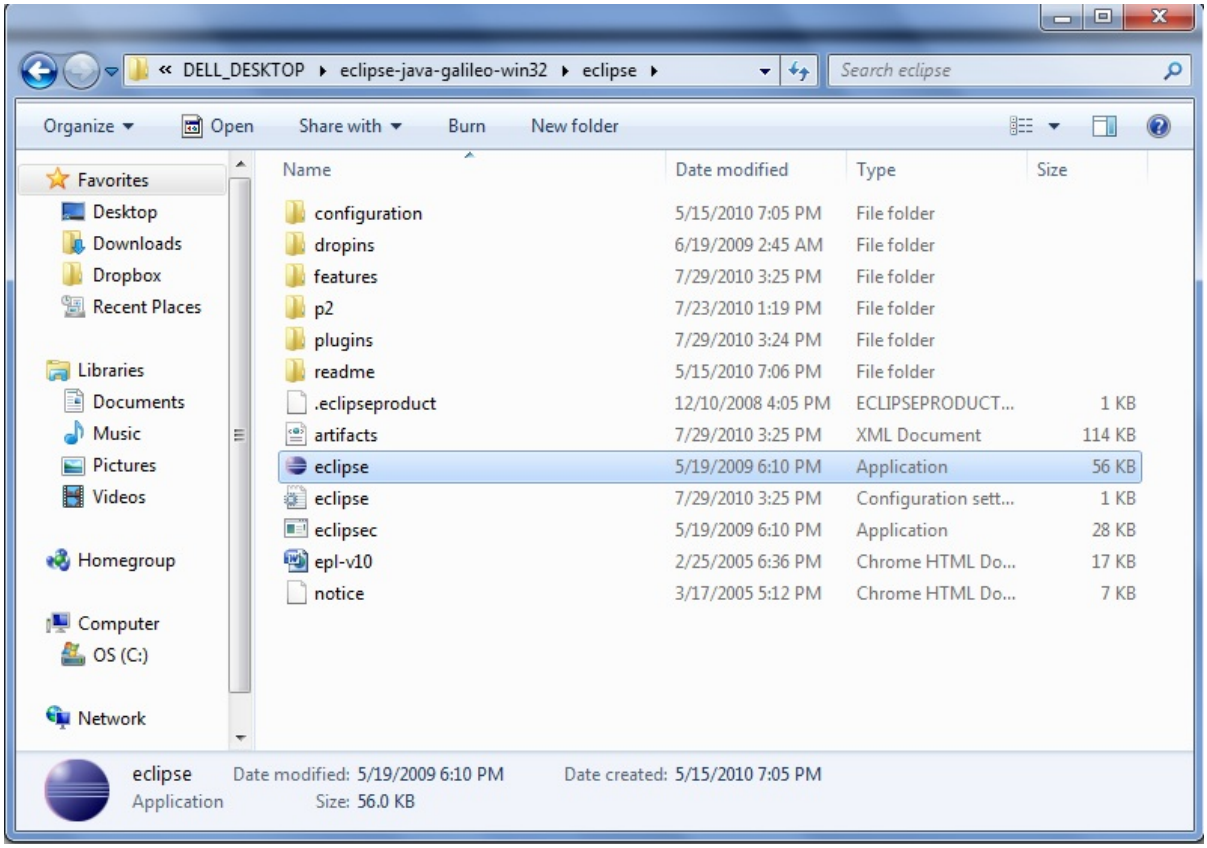

Figure 6.1. Starting Eclipse by double clicking on Eclipse icon.

Figure 6.2. Opening the Eclipse workbench.

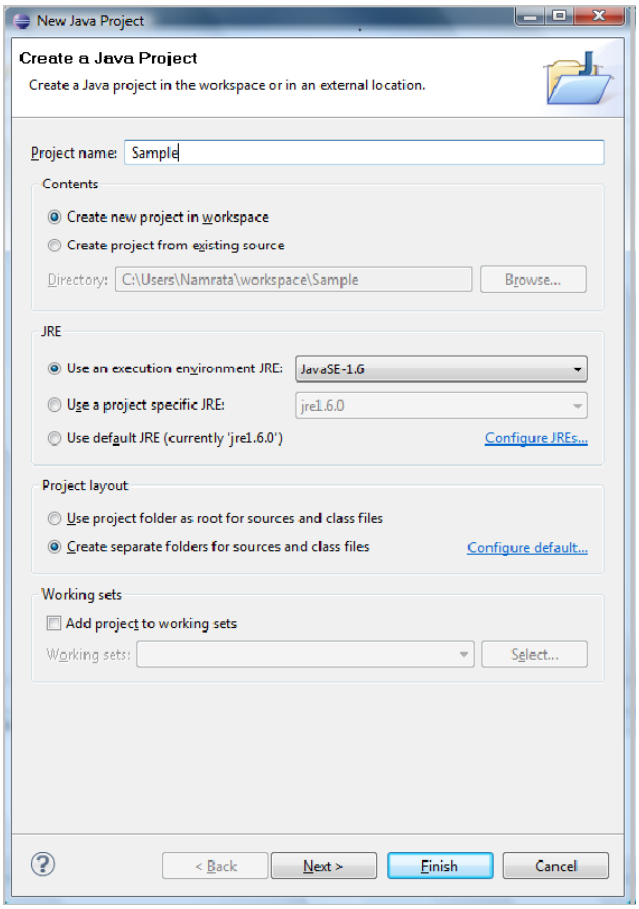

Figure 6.3. Creating a new Java project.

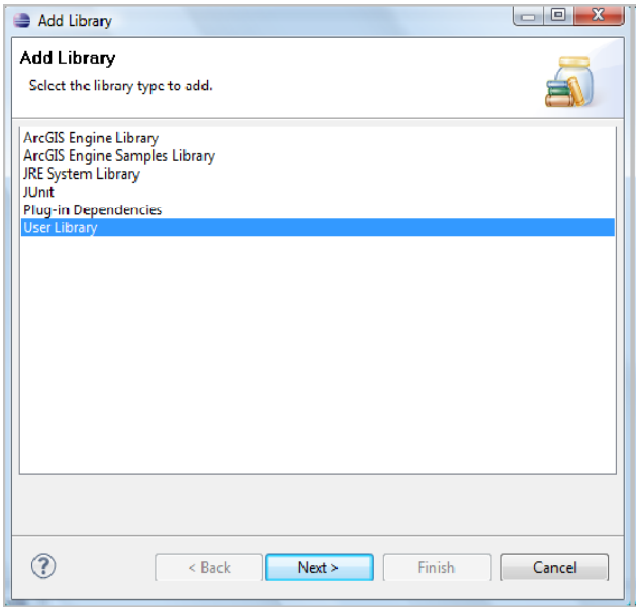

Figure 6.4. Selecting user library.

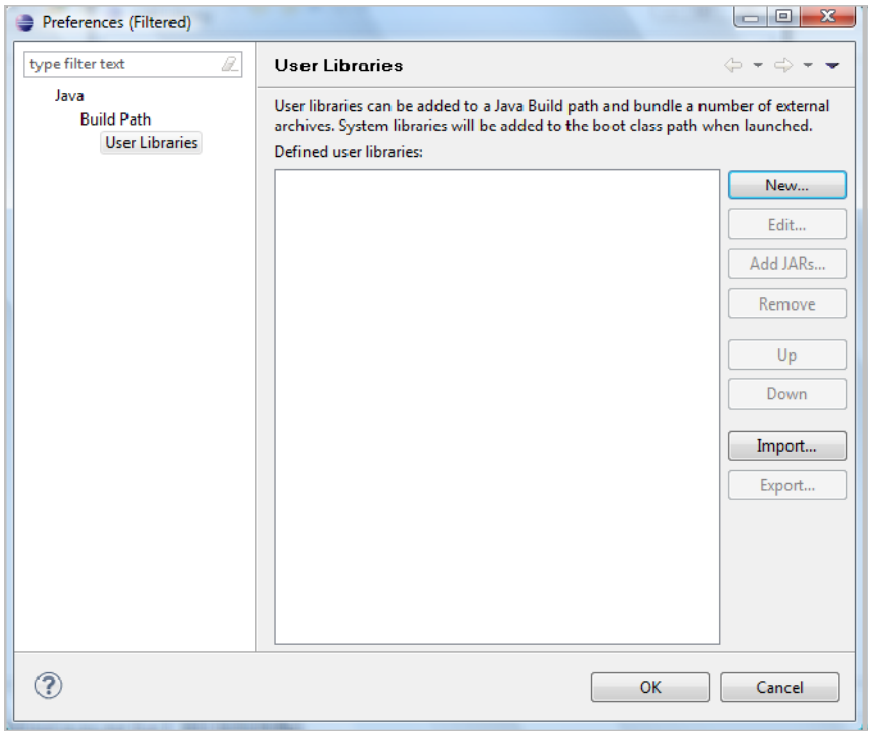

Figure 6.5. Creating a new library.

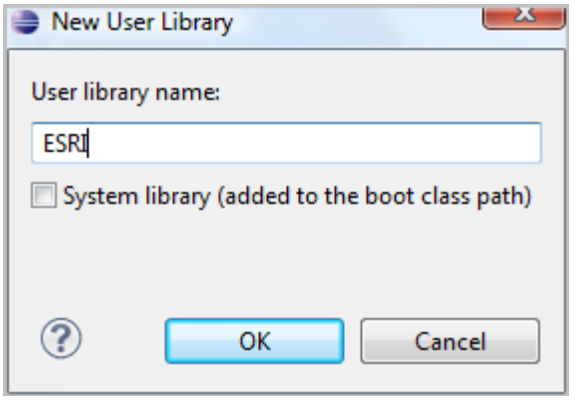

Figure 6.6. Adding library name.

15. Now attach to attach JavaDocs to the jar files expand one of the jar files and click on JavaDoc location and then click edit. Figure 6.9 shows the step execution.
16. In the JavaDoc location path, navigate to the JavaDoc folder in the ESRI documentation directory. Figure 6.10 shows the process.
17. Click ok and do the same for other jar files.

Note that use of an IDE is not completely without drawbacks. The IDE generates considerable code automatically, without the programmer paying much attention to the exact nature of this code and how it works.

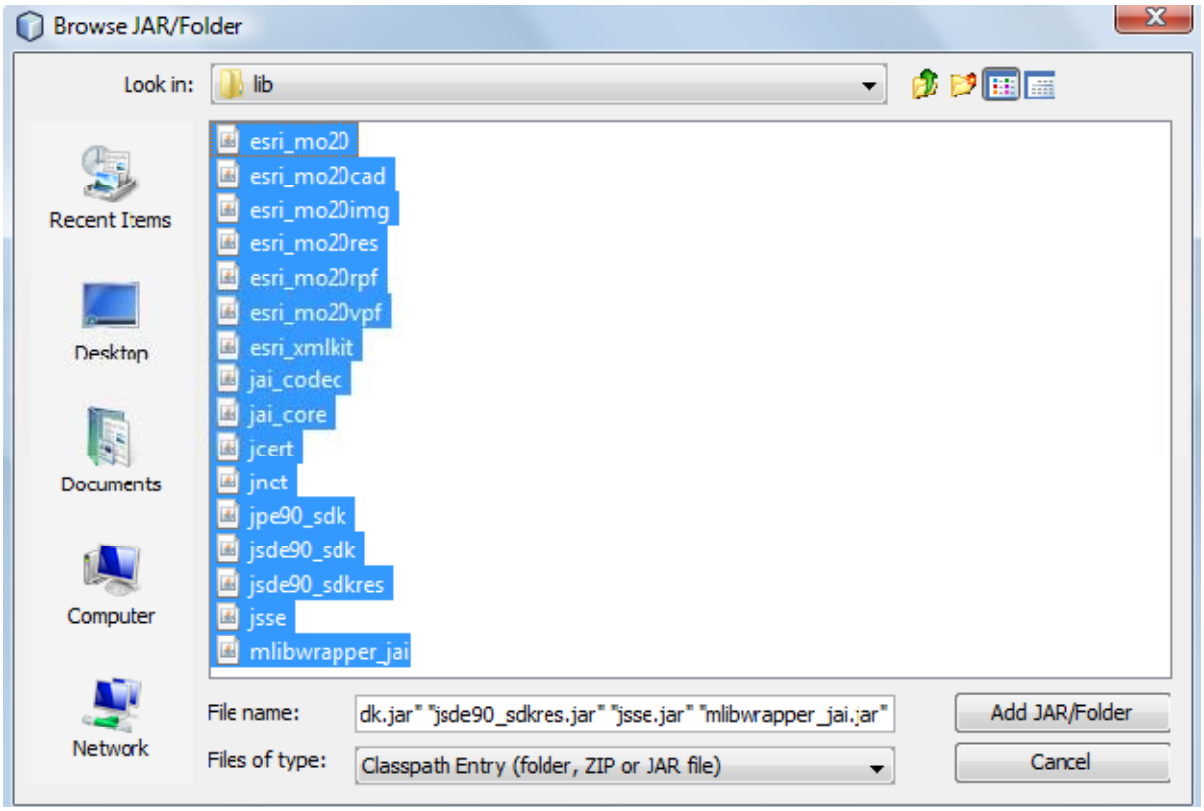

Figure 6.7. Selecting JAR files.

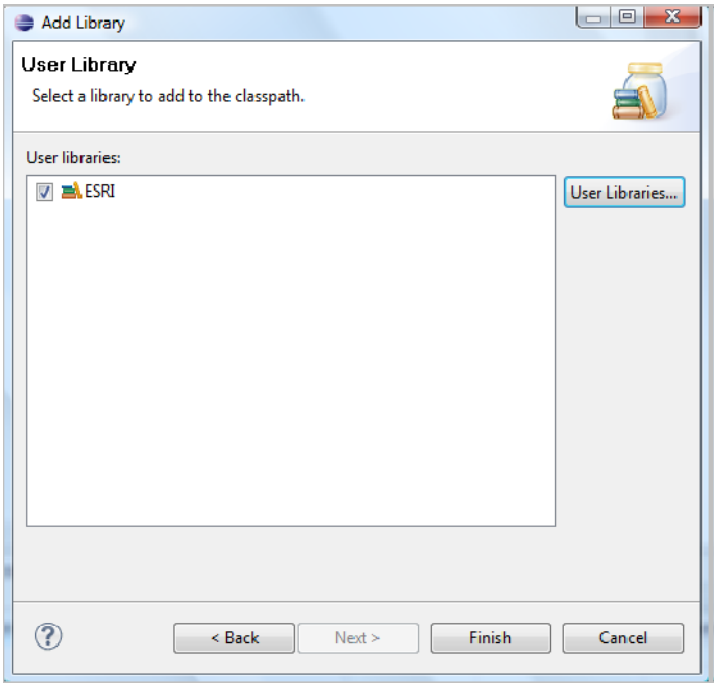

Figure 6.8. Make sure the new library checkbox is selected.

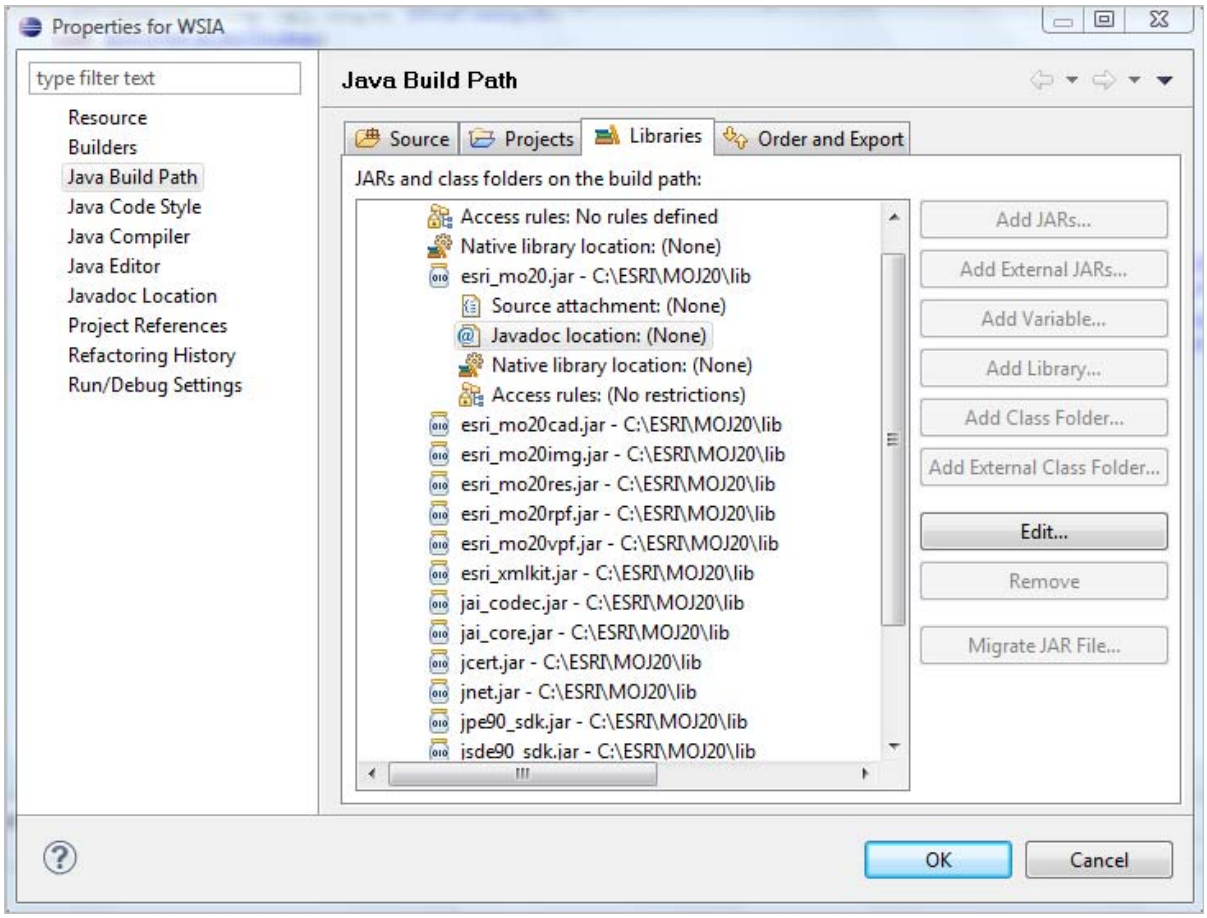

Figure 6.9. Specify the JavaDoc location.

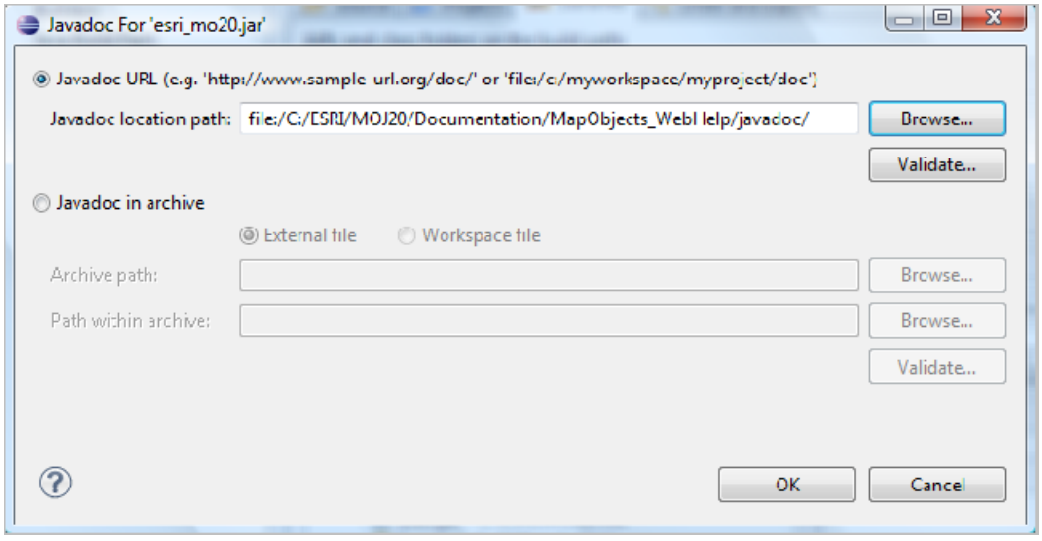

Figure 6.10. Attaching JavaDoc to the JAR files for references in the JAVA editor.

CHAPTER 7

MAPOBJECTS TOOLBAR

Many default beans are provided by MapObjects Java Edition which the user can add to the application and can be dragged and dropped onto the Eclipse workspace to provide basic functionality. Several toolbars are provided by MapObjects which can provide commonly used functions such as open project, save project etc. Refer to the topics in Figure 7.1 to understand using tools of the map view.

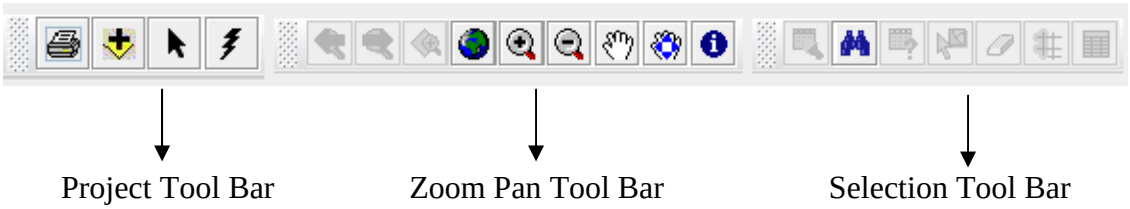

Figure 7.1. MapObjects toolbar.

7.1 PROJECT TOOLBAR

The Project Tool Bar allows user to add a layer, print a map image to file etc. The actions performed by the various options in the project tool bar are as follows [3]:

- Print Map – Use this option to print the map for the selected layer(s) in the TOC.
- Add Layer – Use this option to add layer in the TOC which shows different map.
- Arrow – This is use to resume the cursor from previous selection.
- Hotlink – Customized option used for representing detailed information of the highlighted selection of the layer.

  This is a custom toolbar not provided by MOJO.

7.2 ZOOMPANTOOLBAR

The ZoomPanToolbar allows an end-user to change the map visual look in a variety of ways. By these tools more detailed and zoomed views of desired location can be viewed. The actions performed by the various options in the zoom pan tool bar are as follows:

- Go Previous - Zooms to previous extent stored in extent history.
- Go Next - Zooms to next extent stored in extent history.

- ZoomToSelected - Zooms the map to all selected features in selected layer.
- ZoomToFullExtent - Zooms to extent all layers within map.
- Zoom In - Helps in zooming in. This can be done by selecting the tool and then dragging to sketch a rectangle on the area you want to Zoom In.
- Zoom Out - Helps in zooming out. This can be done by selecting the tool and then dragging to sketch a rectangle on the area you want to zoom out.
- Pan - The Pan option allows the user to drag the map to have better visibility in case the map is in zoomed mode to have better visibility.
- PanOneDirection - This tool allows the user to pan in 1 of the 4 directions allowed: East, West, North and South.
- Identify - Performs an Identify on the active layer.

7.3 SELECTION TOOLBAR

The Selection Toolbar allows the user to perform feature selection based upon attribute or spatial queries. It helps the users to create queries on the selected layer. The actions performed by the various options in the zoom pan tool bar are as follows:

- Find - This option is used for locating features whose attributes contain an end-user provided string.
- Search - This option opens up a dialog for locating features based on a predefined “stored query”.
- Query Builder - This option opens up a dialog based on a query that an end-user constructs.
- Select Feature - This option allows to select a specified area selected by different means as rectangle, circle, line and polygon.
- Clear Selection - This option allows removing the selected features on the active layer.
- Buffer - This option opens a dialog for constructing a polygon around the currently selected feature.
- Attributes - This option displays the attributes of the layer selected on the TOC.

CHAPTER 8

CUSTOM TOOLBAR

In addition to the toolbars provided by MapObjects, custom toolbars can be created in Java to enhance the usability of the GIS tool (Table 8.1). One such toolbar is used in this application.

Table 8.1. Legend of Custom Toolbar

Number	Tool Name	Class or Function Name
1	Print	com.esri.mo2.ui.bean.Print
2	Add Layer	com.esri.mo2.ui.tb.LayerToolBar
3	Delete Layer	com.esri.mo2.map.dpy.Layer
4	Pointer	Arrow.java

PRINT: This tool allows users to print map an attribute table (see Figure 8.1). The following is the code snippet which shows how this tool has been implemented:

```
ActionListener lis = new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
        Object source= ae.getSource();
if(src == print_button)
{
        com.esri.mo2.ui.bean.Print mapPrint = new com.esri.mo2.ui.bean.Print();
        mapPrint.setMap(map);
        mapPrint.doPrint();
}
```

ADD LAYER: This tool allows user to add layers to the table of contents (TOC). The setMap() function must be called before adding. Figure 8.2 shows how the layer can be added to the map. For adding a layer to the map click on the plus symbol on the tool bar; it will open a browser window; then we can browse to the specific location to add the layer.

The following code [5] to integrate the toolbar with the GIS tool:

```
static com.esri.mo2.ui.bean.Map map = new com.esri.mo2.ui.bean.Map();
AddLyrDialog aldlg = new AddLyrDialog();
aldlg.setMap(map);
aldlg.setVisible(true);getContentPane().add(addlg BorderLayout.NORTH);
```

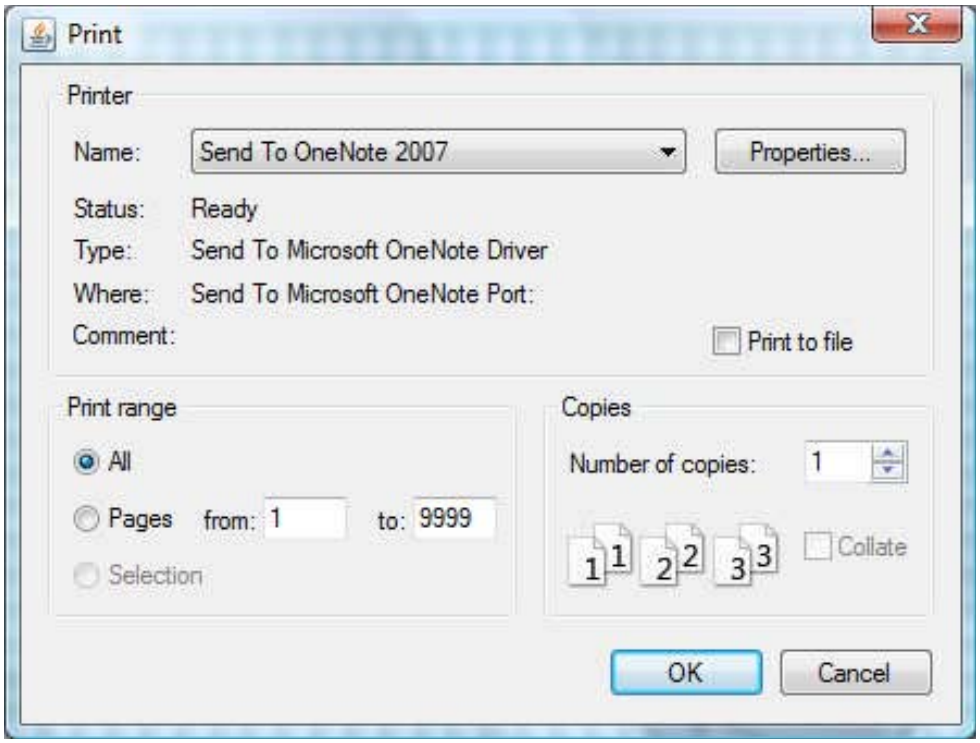

Figure 8.1. Print dialog.

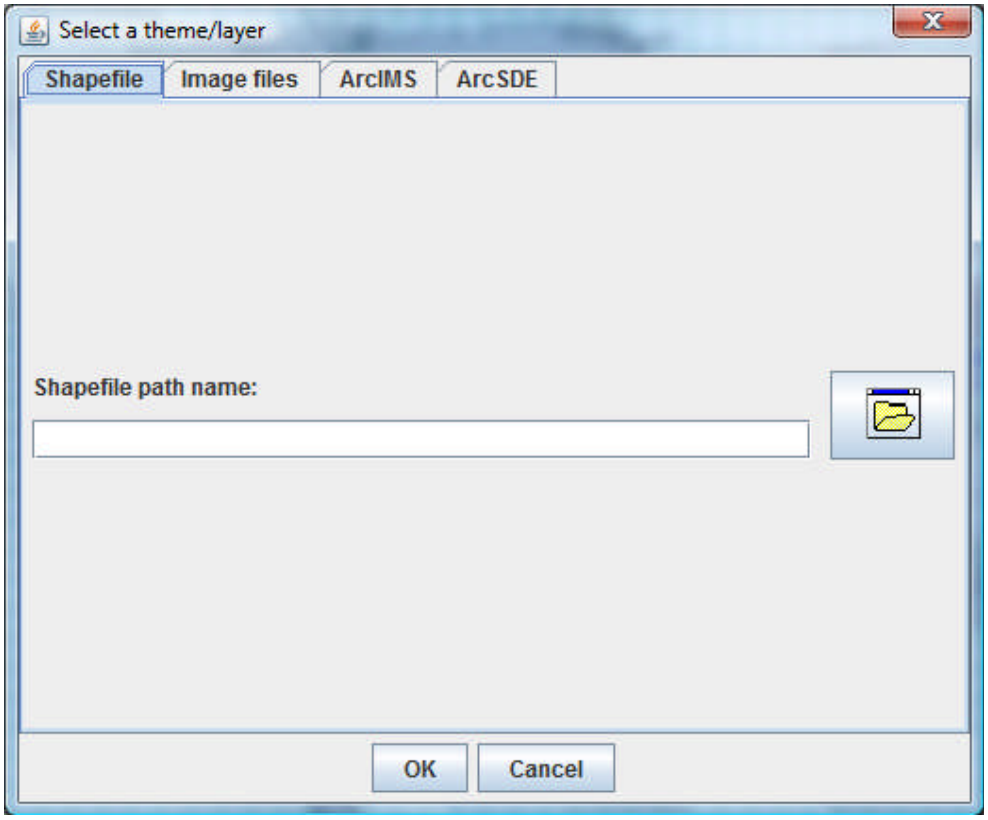

Figure 8.2. Add layer dialog window.

DELETE LAYER: This toolbar provides functionality to allow the user to delete layers from the table of contents. The same functionality is also available through menu. The following is the code snippet:

```
com.esri.mo2.map.dpy.Layer dpylayer = legend.getLayer();
map.getLayerset().removeLayer(dpylayer);
```

ARROW: If some tool has selected before, and the user wants to go to "no tool selected", they can use the arrow tool which is actually a no selection tool. The following is the code:

```
snippet to show how this functioanality is implemented.
Arrow arrow = new Arrow(); \\ creating the object of Arrow class
button_lis = new ActionListener() {
public void actionPerformed(ActionEvent ae) {
else if (src == arrow_button) {
        setCursor(java.awt.Cursor.DEFAULT_CURSOR);
        map.setSelectedTool(arrow); \\ setting the map tool to arrow
        }
}
```

The following code explains how to create the Arrow class:

```
import com.esri.mo2.ui.bean.Tool;
import java.awt.event.MouseEvent;
public class Arrow extends Tool {
        public void mouseClicked(MouseEvent me){
        }
}
```

CHAPTER 9

ADDITIONAL TOOLS

This GIS multimedia teaching tool about Africa provides other features in addition to the default GIS toolbars. These tools enhance the functionality and usability for the user. These tools provide better understanding of data and facility to evaluate the varied data so as to make comparisons and get the collective information of all the categories.

9.1 HOTLINK

The Hotlink tool provides information about the independence history, government type, area, population, capital city, colonization history of each African country as well as points of interests around African continent. When the hotlink toolbar option is clicked, a window pops up giving the above information for each African country. To use this tool, select a layer (africa1 or africapoi) from the TOC i.e. left panel of the tool, and click on the hotlink button. Now click on any of the African country of the feature layer on the map. A window opens up showing the information of that country. The “government type” displayed in the hotlink popup should ideally be updated frequently, and was obtained from the web and Wikipedia country by country. Figure 9.1 shows the popup window which shows the data of the selected country. This code snippet shows that the most recent clicked shapefile is used and data is accessed from its attribute table columns and displayed on the window:

```
FeatureLayer flayer2 = (FeatureLayer) pe.getLayer();
com.esri.mo2.data.feat.Cursor c = pe.getCursor();
Feature f = null;
Fields fields = null;
if (c != null) f = (Feature)c.next();
fields = f.getFields();
if(flayer2.getName().equals(“afrida1”)){
        mygdi = f.getValue(11).toString();
        mypop = f.getValue(13).toString();
} elseif(flayer2.getName().equals(“africapoi”)){
        myaidsD = f.getValue(11).toString();
        myaidsL=f.getValue(12).toString();
}
```

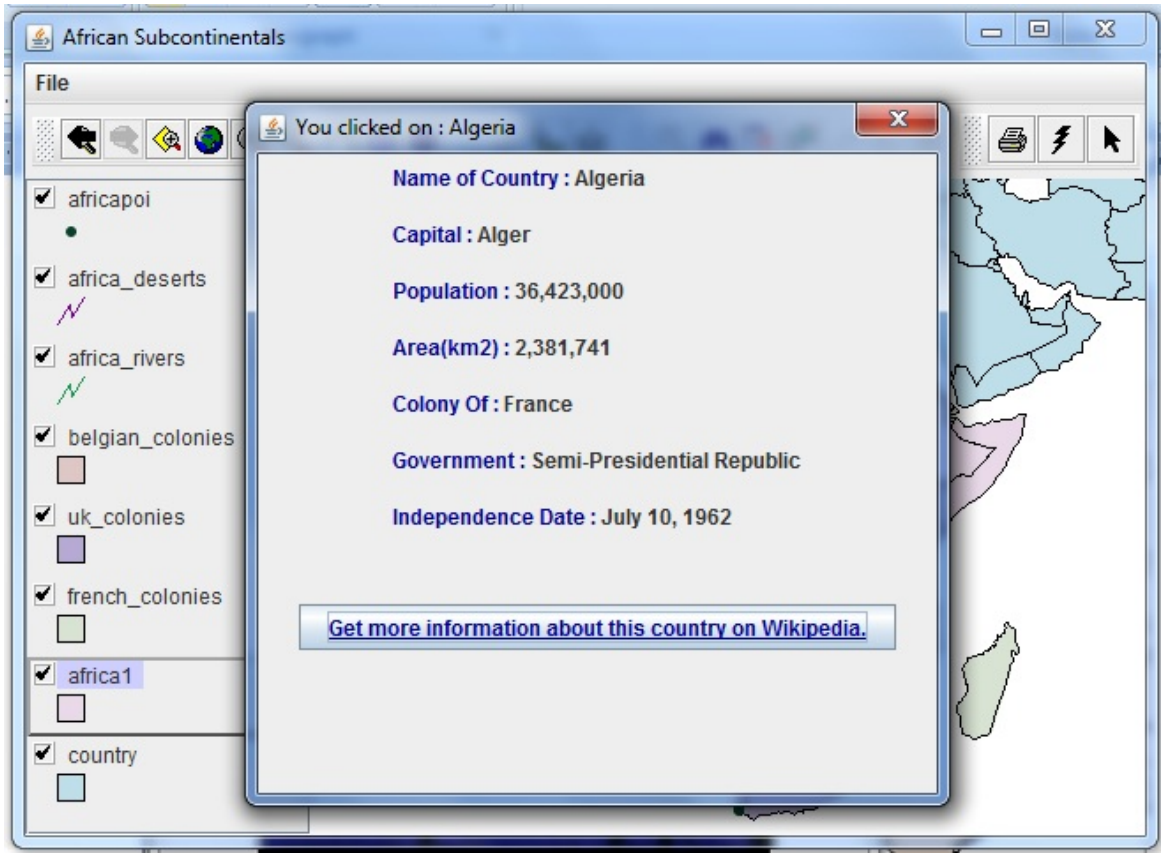

Figure 9.1. GIS tool with Hotlink window.

9.2 LEGEND EDITOR

GIS toolkit offers one more important tool which helps users to compare the data and is known as Layer Properties tool. This tool allows users to change the properties of a layer such as its renderer and symbols by a user interface. The layer properties tool has been implemented using the com.esri.mo2.ui.ren.LayerProperties class [5].

The Legend Editor tool is accessed through the File Menu. Figure 9.2 shows the Layer Properties window. It has three tabs: Symbols, Labels and Generals. In the symbol tab we get option to select the single symbol, graduated symbol and unique symbol. By using these symbols the layer feature can be represented. By using label tab we can provide labels to the layer feature and it can be labeled by different categories stored in the attribute table. By using general tab, users can select the option of when the feature layer should be shown.

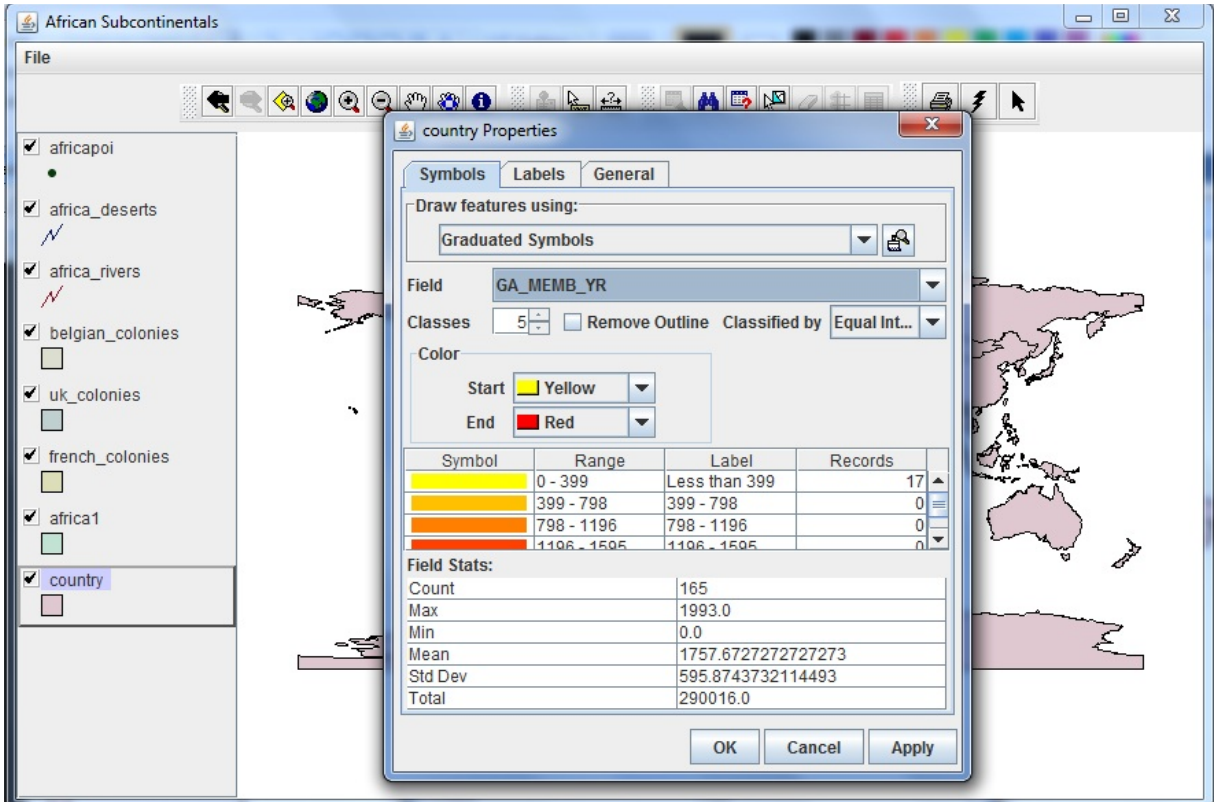

Figure 9.2. Layer properties window.

It can be shown at all times or it can be shown only when the layer is in the range mentioned by the user. In Figure 9.2 symbol tab is used and graduated symbol is selected.

Figure 9.3 illustrates the use of graduated symbols. Applying the graduated symbols feature on a given field divides the data into intervals and gives it shades from light to dark.

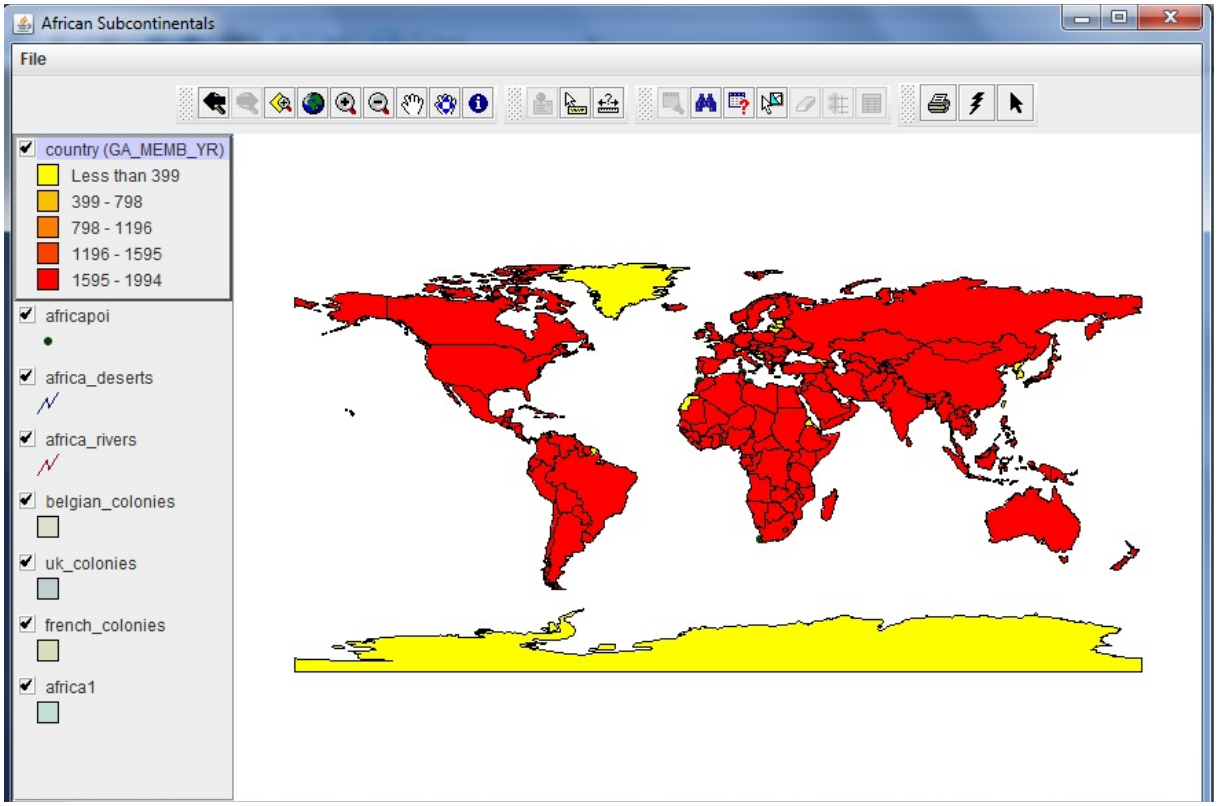

Figure 9.3. GIS tool with graduated symbols on the layer using legend editor.

CHAPTER 10

SCREENSHOTS

This chapter contains the screenshots of this tool. Some of the screenshots have already been covered in previous chapters. The remaining screenshots are presented here in Figures 10.1 to 10.9.

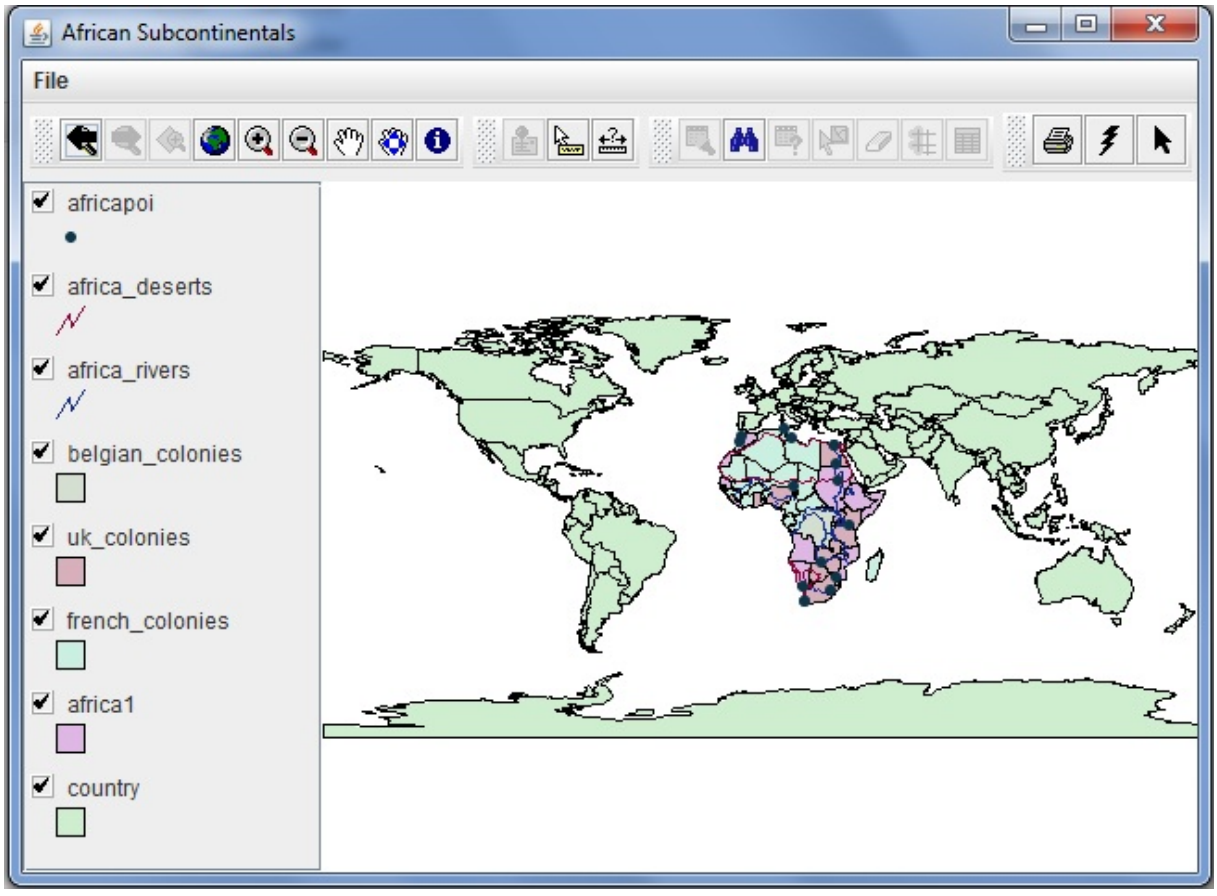

Figure 10.1. GIS multimedia teaching tool about Africa start screen.

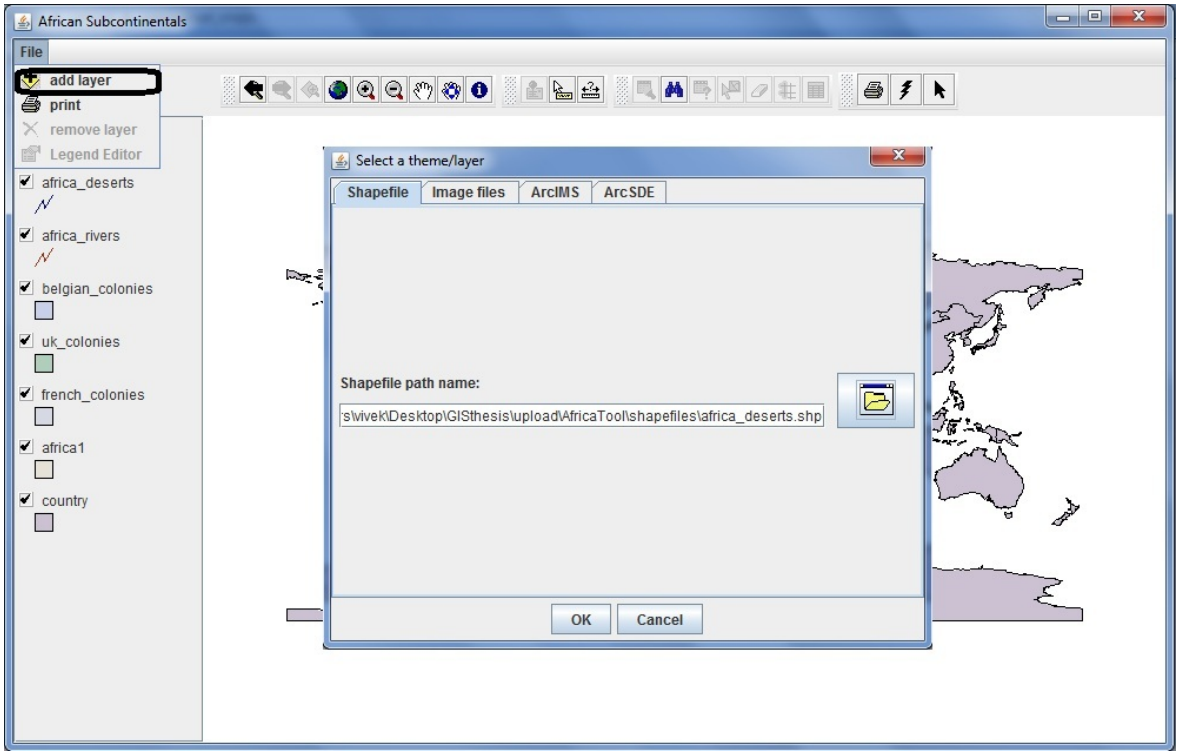

Figure 10.2. Add layer.

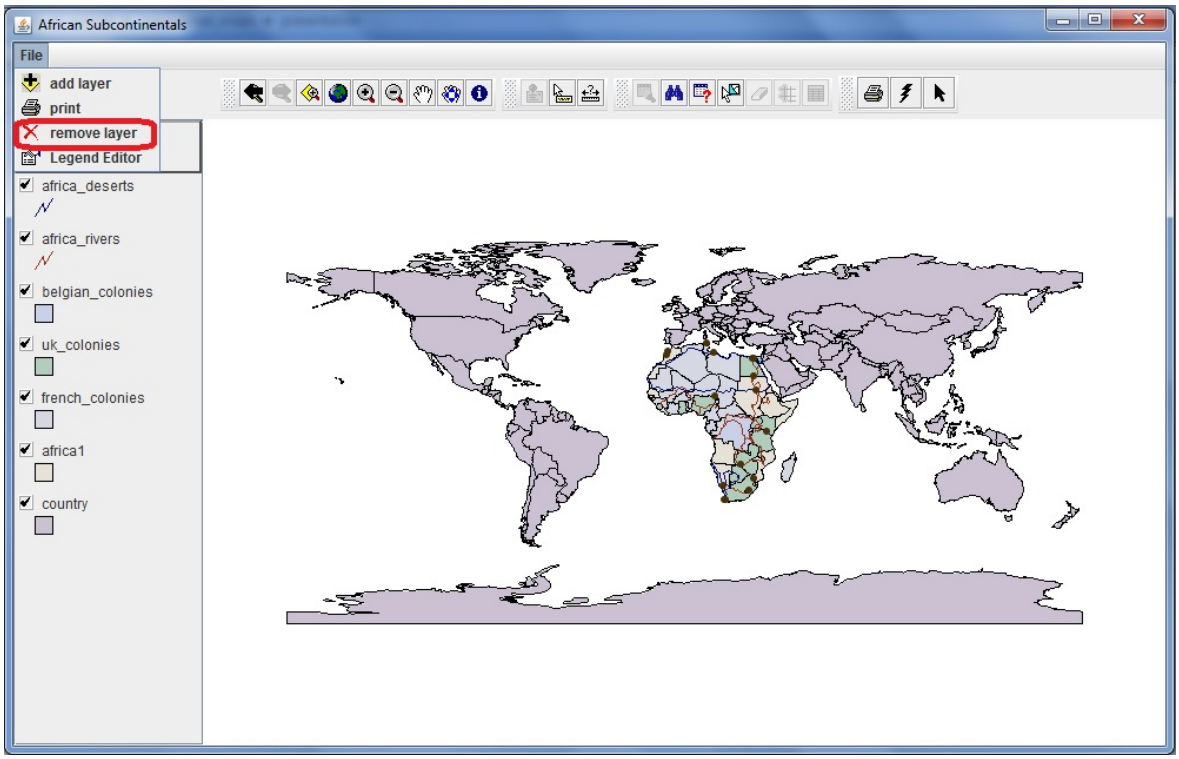

Figure 10.3. Remove layer.

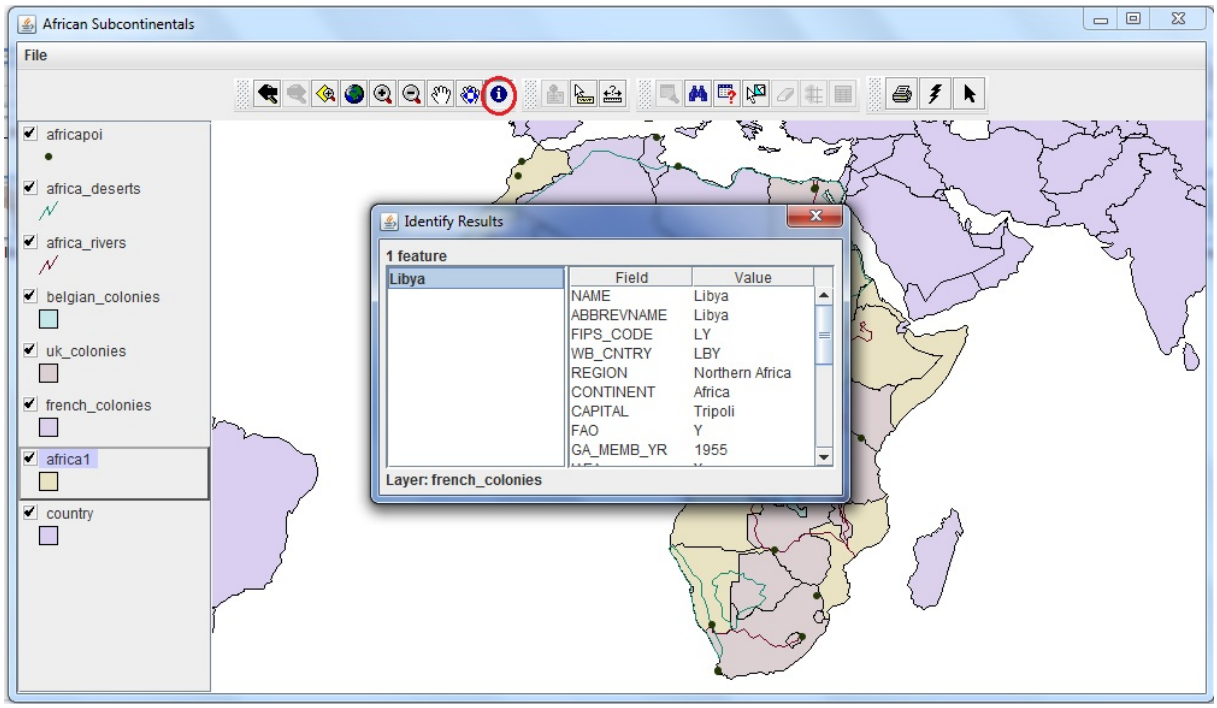

Figure 10.4. Identify tool.

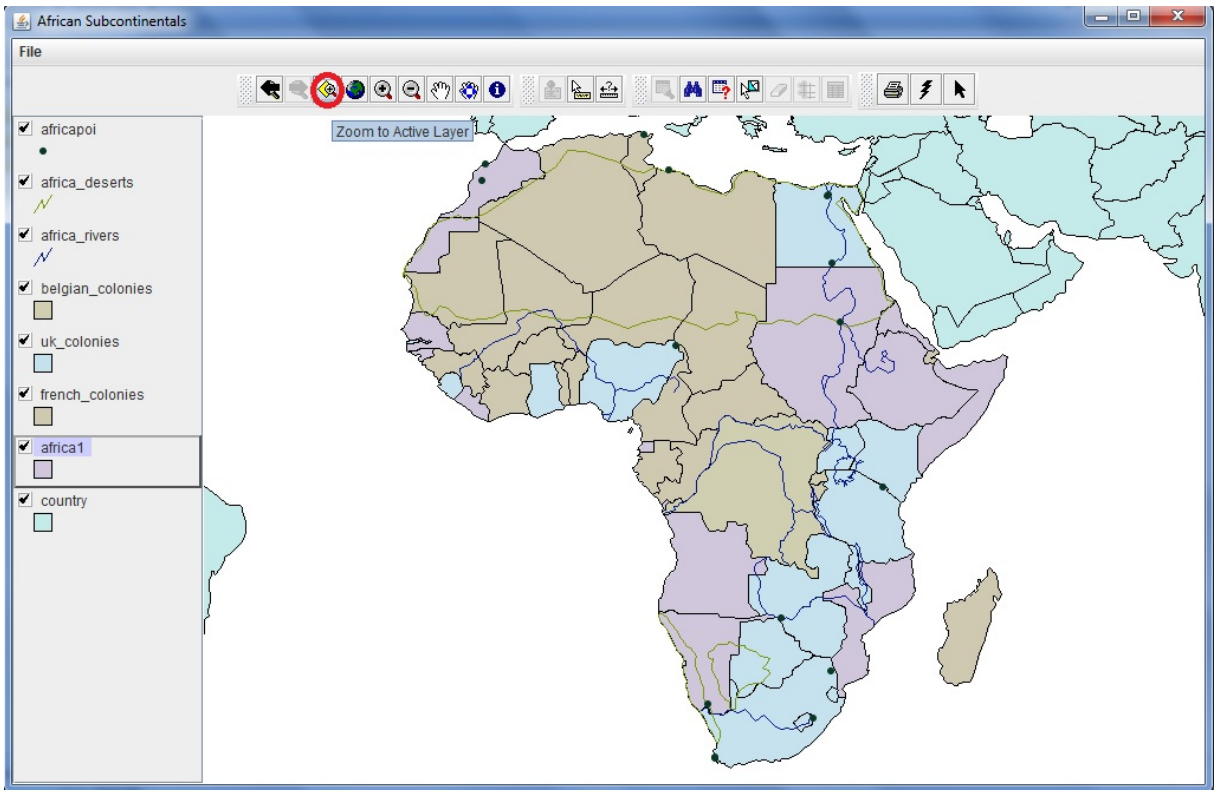

Figure 10.5. ZoomToActiveLayer.

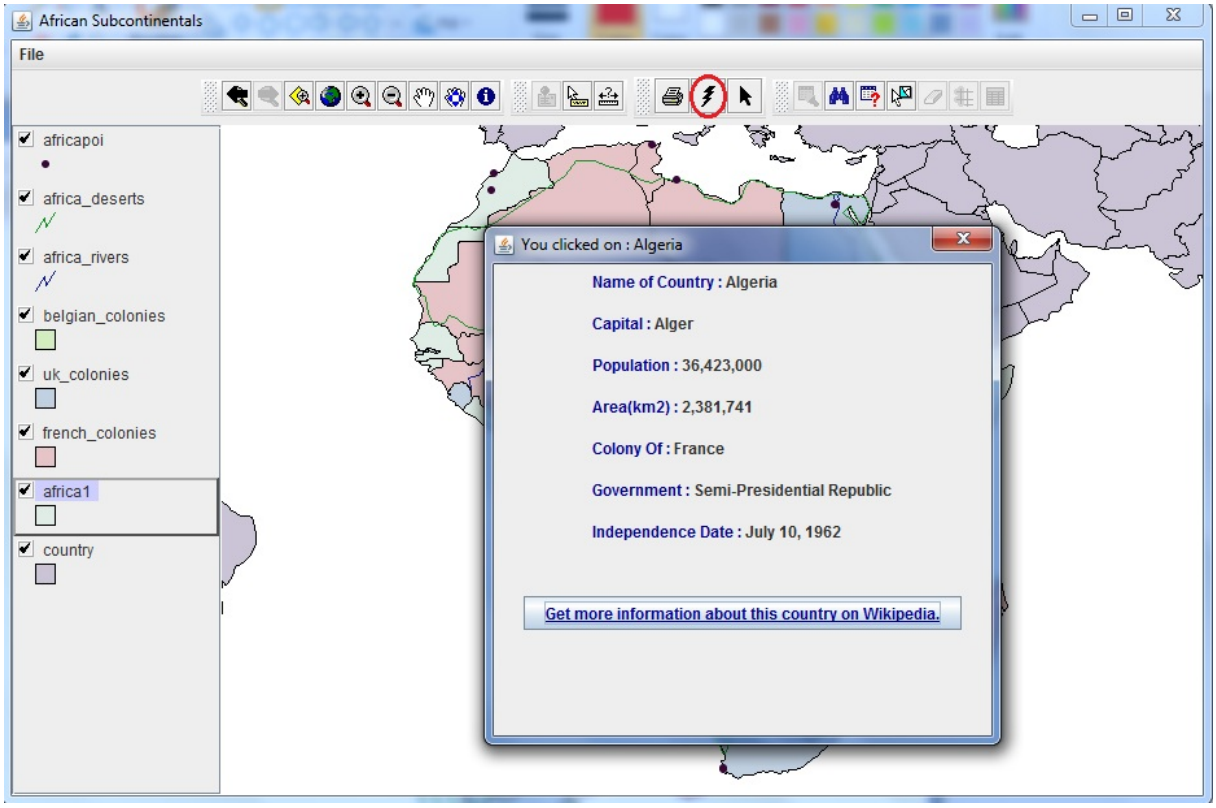

Figure 10.6. Hotlink tool applied on africa1 layer.

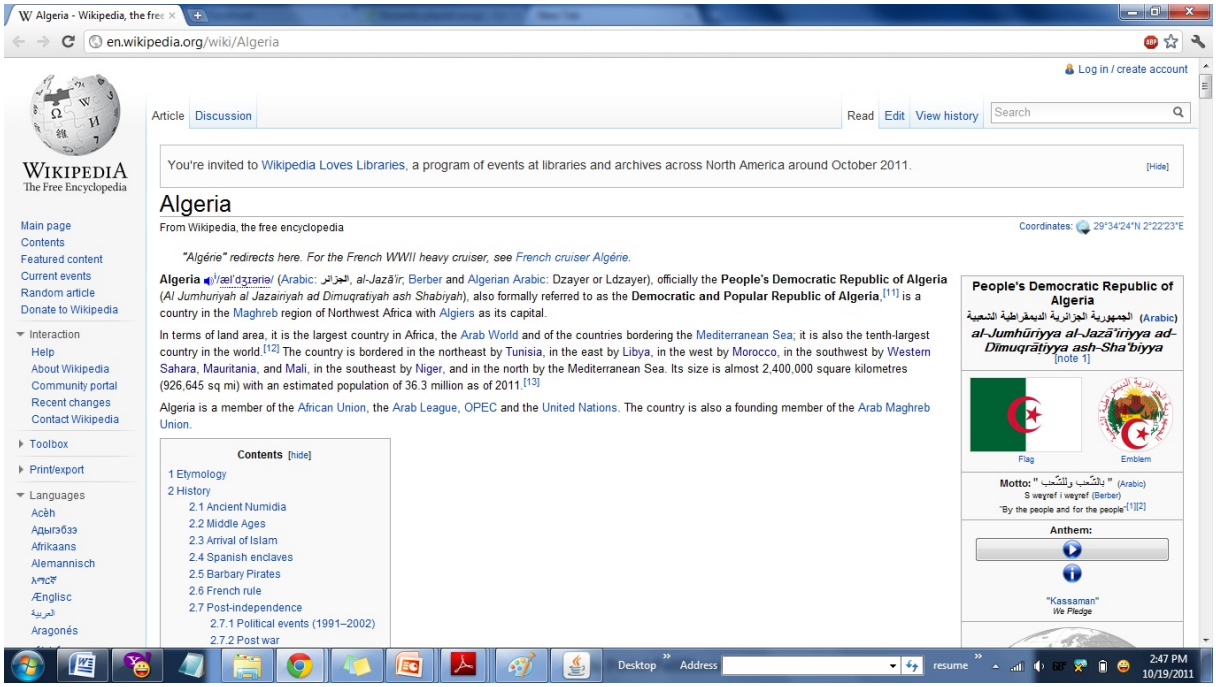

Figure 10.7. Wikipedia page of the country selected by Hotlink tool.

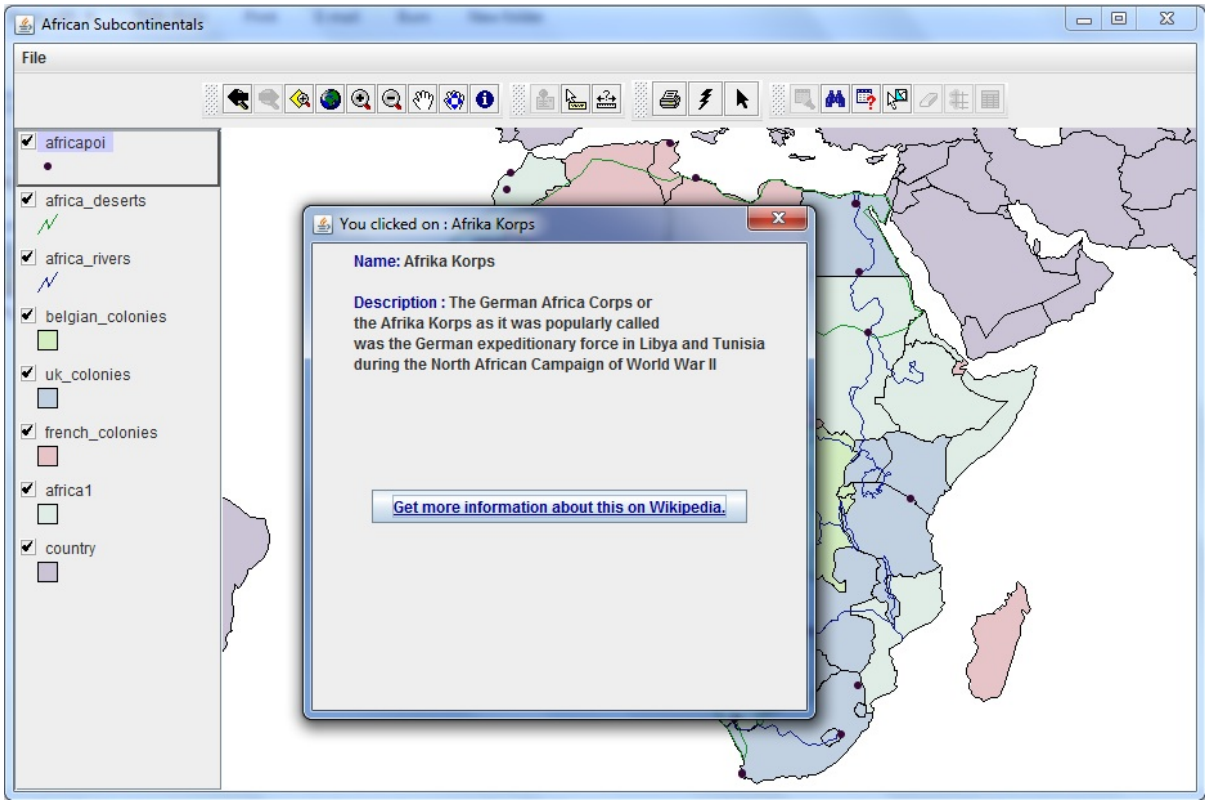

Figure 10.8. Hotlink tool applied on africapoi layer.

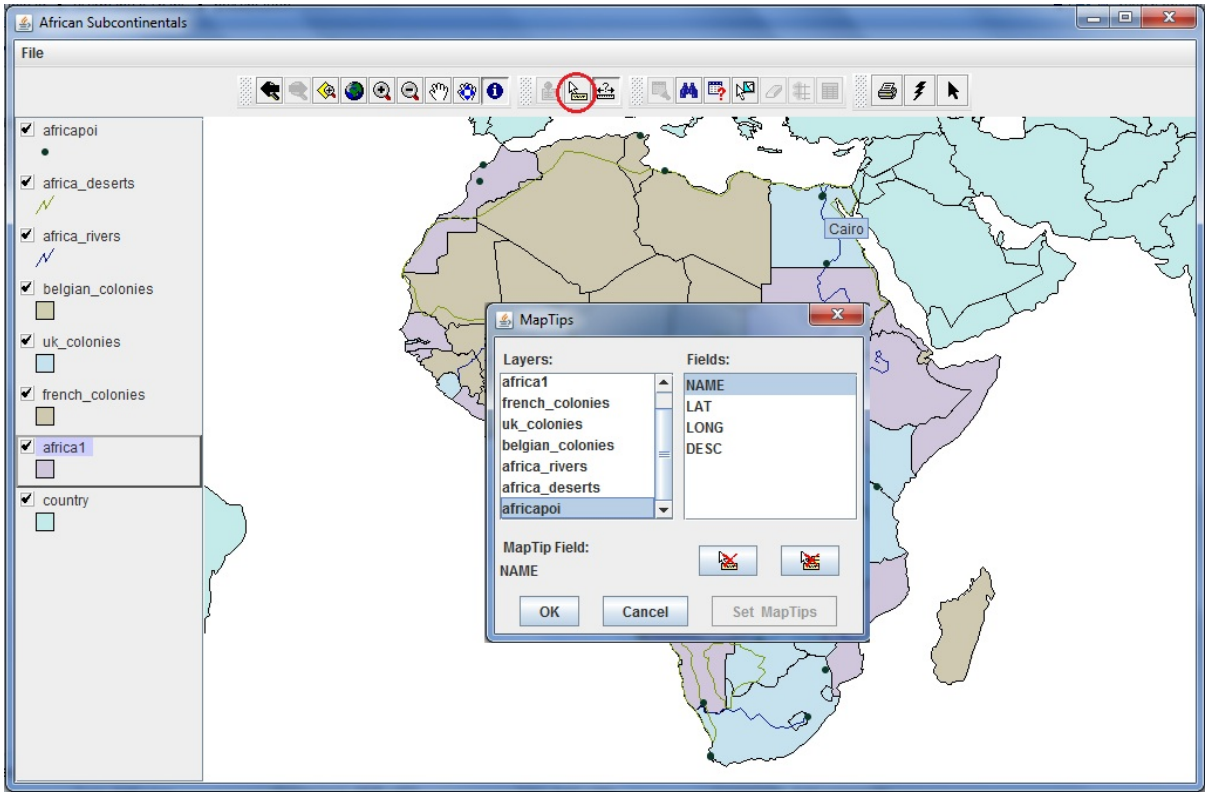

Figure 10.9. Map tips feature applied on NAME field africapoi layer.

CHAPTER 11

CONCLUSION AND OBSTACLES

Africa continues to be often ignored in university classes in departments like history and geography, so an application of this kind is more valuable, if people can be shown to it.

This played a role in the difficulty of finding both information, e.g. current type of government, and feedback from expert.

MOJO provides easy extendibility. It will help to add new features or functionality to this tool effortlessly. This tool can be converted to a web based application to provide better accessibility to the users.

CHAPTER 12

FUTURE ENHANCEMENTS

This tool provides a great framework which can be expanded to add more features and functionalities. This tool has been developed using object oriented design in java which will allow reusing classes, and extra features can be added with minimal code changes.

Some of the future enhancements proposed are:

- Deploying the application on the web as browser hosted applet, so that it can be used without installing it on each machine.
- This tool can be localized into different languages.
- Application can be made for mobile devices and tablets.
- More customized tools can be added as per the requirement.
- More countries can be added.
- Easy way to update time sensitive material such as type of government.

BIBLIOGRAPHY

[1] Oracle. Java SE Downloads, 2011. http://java.sun.com/javase/downloads/index.jsp, accessed Oct. 2011.

[2] Wikipedia. Java Programming Language, 2011. http://en.wikipedia.org/wiki/Java_(programming_language), accessed Oct. 2011.

[3] Sakshi Arora. GIS and multimedia presentation of women's status across the world. Master's thesis, San Diego State University, San Diego, CA, 2011.

[4] WEB-DOT-DEV. Java Advantages and Disadvantages, 2011. http://www.webdotdev.com/nvd/articles-reviews/java/java-advantages-and-disadvantages-1042.html/, accessed Oct. 2011.

[5] Carl Eckberg. Notes on MapObjects Java Edition, CS537 handout. San Diego State University, San Diego, CA, 2011.

[6] CMC Government Services. MapObjects Java Edition, 2011. http://www.cmcgov.com/store/pc/MapObjects-Java-Edition-p34.htm, accessed Sept. 2011.

[7] Namrata Verma. GIS multimedia tool. Master's thesis, San Diego State University, San Diego, CA, 2011.

[8] Wikipedia. Class Diagram, 2011. http://en.wikipedia.org/wiki/Class_diagram, accessed Oct. 2011.